

EF409 652 45545

AUS9-2000-0851

1

Architecture for a Unified Synchronous and Asynchronous Sealed Transaction

Field of the Invention

5

This invention relates to transmitting information over a global computer network and in particular to an architecture and method for transmitting sealed information across multiple synchronous and asynchronous protocols over the global computer network.

10

Background of the Invention

The expansion of computer networks and the advancements in information technology have attached a growing number of persons and companies to conduct business over the computing networks. This practice has substantially increased the number of message transmissions. These message transmissions occur over communication mediums use one of two communication protocols: synchronous or asynchronous. Synchronous communication protocols are seen in point-to-point communications. These transactions involve at least two entities in direct contact with each other such as a telephone conversation or facsimile transmission. These transactions occur over a secure line with minimum risk of compromising any confidential information that may be transmitted during the transaction. One example of a synchronous protocol is Secure Sockets Layer (SSL). The other communication protocol type is asynchronous. With this protocol type, there is no direct connection between entities. Transmissions over a global computer network such as the Internet are not point-to-point. In fact, a message may have traveled through several entities before reaching its final destination. An example of this protocol is a typical electronic mail transmission. These communication protocols use encryption techniques and other procedures to ensure the security, authentication and integrity the of message transmissions. One area of encryption techniques use keys to authenticate and validate transmitted messages.

One type of encryption, Symmetric, or private key, encryption (also known as conventional encryption) is based on a secret key that is shared by both communicating parties. The sending party uses the secret key as part of the mathematical operation to

[illegible]

AUS9-2000-0851

2

encrypt (or encipher) plaintext to ciphertext. The receiving party uses the same secret key to decrypt (or decipher) the ciphertext to plaintext. Examples of symmetric encryption schemes are the RSA RC4 algorithm (which provides the basis for Microsoft Point-to-Point Encryption (MPPE), Data Encryption Standard (DES), the International Data Encryption Algorithm (IDEA), and the Skipjack encryption technology proposed by the United States government (and implemented in the Clipper chip).

With symmetric encryption, both the sender and receiver have a shared secret key. The distribution of the secret key must occur (with adequate protection) prior to any encrypted communication. However, with asymmetric encryption, the sender uses the recipient's public key to encrypt or digitally sign messages, while the receiver uses their key to decipher these messages. The public key can be freely distributed to anyone who needs to encrypt messages to the owner of the public key or to verify digitally signed messages by the private key that corresponds to the public key. The owner of the key pair only needs to carefully protect the private key.

Another encryption technique, asymmetric or public key encryption uses two different keys for each user: one key is a private key known only to the user to which the key pair belongs; the other is a corresponding public key, which is accessible to anyone. The encryption algorithm mathematically relates the private and public keys. One key is used for encryption and the other for decryption, depending on the nature of the communication service being implemented. The public key is made public by distributing it widely. The private key is never distributed; it is always kept secret. In addition, public key encryption technologies allow digital signatures to be placed on messages. A digital signature uses the sender's private key to encrypt some portion of the message. When the message is received, the receiver uses the sender's public key to decipher the digital signature as a way to verify the sender's identity and the integrity of the message.

To secure the integrity of the public key, the public key is published with a certificate. A certificate (or public key certificate) is a data structure that is digitally signed by a certificate authority (CA). The CA is an authority that users of the certificate can trust. The certificate contains a series of values, such as the certificate name and usage, information identifying the owner of the public key, the public key itself, an

[illegible]

AUS9-2000-0851

3

expiration date, and the name of the certificate authority. The CA uses its private key to sign the certificate. If the receiver knows the public key of the certificate authority, the receiver can verify that the certificate is indeed from the trusted CA, and therefore contains reliable information and a valid public key. Certificates can be distributed
5 electronically (via Web access or e-mail), on smart cards, or in an LDAP database. Public key certificates provide a convenient, reliable method for verifying the identity of a sender. IPSec can optionally use this method for end-to-end authentication.

A public and private key pair is a unique association of key values wherein one key can encrypt information and the other can decrypt. For example, the public key can
10 encrypt data and only the corresponding private key can decrypt the data. Public and private keys are used for signing and sending encrypted messages. A public key is typically made available to users on a global computer network (the Internet) within a certificate stored in a publicly accessible Lightweight Directory Application Protocol (LDAP) directory. The associated private key is kept in confidence by the entity, such as
15 the person or cooperation that owns the key pair.

Although a message is encrypted to make it private, there is still a concern that someone might modify the original message or substitute it with a different message. One way to guaranteeing the integrity of the original message is to create a concise summary of the message and send this summary along with the message to the receiving
20 party. Upon receipt of the message, the recipient creates its own summary and compares it with the one sent. If the message summaries agree, then the message was received intact. This message summary is called a message digest, or one-way hash. Message digests are used to create short, fixed-length representations of longer variable-length messages. Digest algorithms are designed to produce unique digests for different
25 messages. Message digest are designed to make it too difficult to determine the main message from the digest and also impossible to find two different messages which create the same digest. This trait of a message digest eliminates the possibility of substituting one message for another while maintaining the same digest.

When one entity sends a message to another entity, the receiving entity needs to
30 ensure that the message is really from the sender, so an intruder does not interfere with or alter that message. A digital signature, created by the sending entity and included with

4

Previous communication protocols have focused on transmission of a sealed transaction between two entities. In a typical SSL transaction such as sending a facsimile message, the destination can perform processing on the information and determine its origin and maintain the originality of the message. The recipient will know what entities were involved in the transaction along the transmission, the actual sender of the message, check to verify that each entity was able to sign that information confirming that they

AUS9-2000-0851

5

were able to take care of the information. An example is purchasing an item over the Internet. This type of transaction is an SSL transaction. However, sending an electronic mail message is an asynchronous transaction. With the expanded use of the Internet and other computing networks, there is a need to be able to have the ability to trace events in the transmission of the message regardless of whether the transmission is synchronous or asynchronous.

Figure 1 consists of 12 histograms arranged in a single column. Each histogram represents the distribution of the number of non-zero elements in the vector x for a specific value of n . The x-axis for all histograms is labeled 'Number of non-zero elements' and ranges from 0 to 120. The y-axis is labeled 'Frequency' and ranges from 0 to 100. The histograms are labeled with n values: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, and 120. As n increases, the distribution of non-zero elements shifts to the right, indicating that more elements in the vector x are non-zero for larger n .

AUS9-2000-0851

6

Summary of the Invention

It is an object of the present invention to provide a communication procedure that can enable secure transmissions across multiple synchronous and asynchronous protocols.

It is another object of the invention to provide a procedure that will enable the recipient of a message to trace all of the events during the message's transmission.

It is another object of the invention to provide a means to store all events of a message transmission.

It is another object of this invention to provide a mechanism that can enable a transaction to span multiple synchronous and asynchronous secure transmissions using public key technologies.

This invention describes a method and architecture to seal a transaction across multiple synchronous and asynchronous protocols over a global computer network using public key and private key technologies. Protocols across global computer networks are available between two entities, such as a client and server, but they are limited in their ability to span multiple entities or hops, while still retaining the identity and original transaction of the initial sender in a standard fashion. In addition to ensuring the security of the transaction, this design addresses a unified transaction that is sealed and can span any combination of synchronous and asynchronous transmissions between multiple entities.

This invention allows any number of entities to participate in the sealed transaction, wherein each entity can add to the transaction, the complete transaction is protected from unintended recipients, and authentication and integrity is ensured with each entity. During the transmission, an entity may receive the transmitted message. The entity may add information or modify information in the message. The changes would be recorded in a data structure called a SignedData object. Each new entity that receives the message during the transmission may add a SignedData object to the transmitted. Through the SignedData objects, at the end of the transmission, there is a complete record of the events that occurred during the transmission of that message. In this method, the authenticity and integrity of the transaction is preserved.

7

Figure 1 depicts data processing equipment a system that can be utilized to implement the present invention;

5 Figure 2 is a diagram of a computer over which messages may be transmitted
synchronously or asynchronously;

Figure 3 is a flow diagram of the steps in the implementation of the message transmission mechanism of the present invention;

Figure 4 is a diagram of a data structure containing information in the transmitted
10 message; and

Figure 5 is a diagram of the signed attribute contained in the data structure.

| Experimental conditions | | Control | | 100% CO ₂ | | 100% N ₂ | | 100% O ₂ | | 100% CO ₂ + 100% N ₂ | | 100% CO ₂ + 100% O ₂ | | 100% N ₂ + 100% O ₂ | |
|-------------------------|-------------|---------|-------------|----------------------|-------------|---------------------|-------------|---------------------|-------------|--|-------------|--|-------------|---|-------------|
| Group | Sample size | Group | Sample size | Group | Sample size | Group | Sample size | Group | Sample size | Group | Sample size | Group | Sample size | Group | Sample size |
| 1 | 10 | 2 | 10 | 3 | 10 | 4 | 10 | 5 | 10 | 6 | 10 | 7 | 10 | 8 | 10 |
| 9 | 10 | 10 | 10 | 11 | 10 | 12 | 10 | 13 | 10 | 14 | 10 | 15 | 10 | 16 | 10 |
| 17 | 10 | 18 | 10 | 19 | 10 | 20 | 10 | 21 | 10 | 22 | 10 | 23 | 10 | 24 | 10 |
| 25 | 10 | 26 | 10 | 27 | 10 | 28 | 10 | 29 | 10 | 30 | 10 | 31 | 10 | 32 | 10 |
| 33 | 10 | 34 | 10 | 35 | 10 | 36 | 10 | 37 | 10 | 38 | 10 | 39 | 10 | 40 | 10 |
| 41 | 10 | 42 | 10 | 43 | 10 | 44 | 10 | 45 | 10 | 46 | 10 | 47 | 10 | 48 | 10 |
| 49 | 10 | 50 | 10 | 51 | 10 | 52 | 10 | 53 | 10 | 54 | 10 | 55 | 10 | 56 | 10 |
| 57 | 10 | 58 | 10 | 59 | 10 | 60 | 10 | 61 | 10 | 62 | 10 | 63 | 10 | 64 | 10 |
| 65 | 10 | 66 | 10 | 67 | 10 | 68 | 10 | 69 | 10 | 70 | 10 | 71 | 10 | 72 | 10 |
| 73 | 10 | 74 | 10 | 75 | 10 | 76 | 10 | 77 | 10 | 78 | 10 | 79 | 10 | 80 | 10 |
| 81 | 10 | 82 | 10 | 83 | 10 | 84 | 10 | 85 | 10 | 86 | 10 | 87 | 10 | 88 | 10 |
| 89 | 10 | 90 | 10 | 91 | 10 | 92 | 10 | 93 | 10 | 94 | 10 | 95 | 10 | 96 | 10 |
| 97 | 10 | 98 | 10 | 99 | 10 | 100 | 10 | 101 | 10 | 102 | 10 | 103 | 10 | 104 | 10 |
| 105 | 10 | 106 | 10 | 107 | 10 | 108 | 10 | 109 | 10 | 110 | 10 | 111 | 10 | 112 | 10 |
| 113 | 10 | 114 | 10 | 115 | 10 | 116 | 10 | 117 | 10 | 118 | 10 | 119 | 10 | 120 | 10 |
| 121 | 10 | 122 | 10 | 123 | 10 | 124 | 10 | 125 | 10 | 126 | 10 | 127 | 10 | 128 | 10 |
| 129 | 10 | 130 | 10 | 131 | 10 | 132 | 10 | 133 | 10 | 134 | 10 | 135 | 10 | 136 | 10 |
| 137 | 10 | 138 | 10 | 139 | 10 | 140 | 10 | 141 | 10 | 142 | 10 | 143 | 10 | 144 | 10 |
| 145 | 10 | 146 | 10 | 147 | 10 | 148 | 10 | 149 | 10 | 150 | 10 | 151 | 10 | 152 | 10 |
| 153 | 10 | 154 | 10 | 155 | 10 | 156 | 10 | 157 | 10 | 158 | 10 | 159 | 10 | 160 | 10 |
| 161 | 10 | 162 | 10 | 163 | 10 | 164 | 10 | 165 | 10 | 166 | 10 | 167 | 10 | 168 | 10 |
| 169 | 10 | 170 | 10 | 171 | 10 | 172 | 10 | 173 | 10 | 174 | 10 | 175 | 10 | 176 | 10 |
| 177 | 10 | 178 | 10 | 179 | 10 | 180 | 10 | 181 | 10 | 182 | 10 | 183 | 10 | 184 | 10 |
| 185 | 10 | 186 | 10 | 187 | 10 | 188 | 10 | 189 | 10 | 190 | 10 | 191 | 10 | 192 | 10 |
| 193 | 10 | 194 | 10 | 195 | 10 | 196 | 10 | 197 | 10 | 198 | 10 | 199 | 10 | 200 | 10 |

AUS9-2000-0851

8

Detailed Description of the Invention

With reference now to Fig. 1, there is depicted a pictorial representation of data processing system 10 which may be used in implementation of the present invention. As may be seen, data processing system 10 includes processor 11 that preferably includes a graphics processor, memory device and central processor (not shown). Coupled to processor 11 is video display 12 which may be implemented utilizing either a color or monochromatic monitor, in a manner well known in the art. Also coupled to processor 11 is keyboard 13. Keyboard 13 preferably comprises a standard computer keyboard, which is coupled to the processor by means of cable 14. Also coupled to processor 11 is a graphical pointing device, such as mouse 15. Mouse 15 is coupled to processor 11, in a manner well known in the art, via cable 16. As is shown, mouse 15 may include left button 17, and right button 18, each of which may be depressed, or "clicked", to provide command and control signals to data processing system 10. While the disclosed embodiment of the present invention utilizes a mouse, those skilled in the art will appreciate that any graphical pointing device such as a light pen or touch sensitive screen may be utilized to implement the method and apparatus of the present invention. Upon reference to the foregoing, those skilled in the art will appreciate that data processing system 10 may be implemented utilizing a personal computer.

The method of the present invention may be implemented in a global computer network environment such as the Internet. With reference now Figure 1b, there is depicted a pictorial representation of a distributed computer network environment 20 in which one may implement the method and system of the present invention. This diagram illustrates the types of components through which sensitive and confidential; voting information may be exposed and the need for extreme security in this voting process. As may be seen, distributed data processing system 20 may include a plurality of networks, such as Local Area Networks (LAN) 21 and 22, each of which preferably includes a plurality of individual computers 23 and 24, respectively. Of course, those skilled in the art will appreciate that a plurality of Intelligent Work Stations (IWS) coupled to a host processor may be utilized for each such network. Any of the processing systems may also be connected to the Internet as shown. As is common in such data processing systems,

AUS9-2000-0851

9

each individual computer may be coupled to a storage device 25 and/or a printer/output device 26. One or more such storage devices 25 may be utilized, in accordance with the method of the present invention, to store the various data objects or documents which may be periodically accessed and processed by a user within distributed data processing system 20, in accordance with the method and system of the present invention. In a manner well known in the prior art, each such data processing procedure or document may be stored within a storage device 25 which is associated with a Resource Manager or Library Service, which is responsible for maintaining and updating all resource objects associated therewith.

Still referring to Fig. 2, it may be seen that distributed data processing system 20 may also include multiple mainframe computers, such as mainframe computer 27, which may be preferably coupled to Local Area Network (LAN) 21 by means of communications link 28. Mainframe computer 27 may also be coupled to a storage device 29 which may serve as remote storage for Local Area Network (LAN) 21. A second Local Area Network (LAN) 22 may be coupled to Local Area Network (LAN) 21 via communications controller 31 and communications link 32 to a gateway server 33. Gateway server 33 is preferably an individual computer or Intelligent Work Station (IWS) which serves to link Local Area Network (LAN) 22 to Local Area Network (LAN) 21. As discussed above with respect to Local Area Network (LAN) 22 and Local Area Network (LAN) 21, a plurality of data processing procedures or documents may be stored within storage device 29 and controlled by mainframe computer 27, as Resource Manager or Library Service for the data processing procedures and documents thus stored. Of course, those skilled in the art will appreciate that mainframe computer 27 may be located a great geographical distance from Local Area Network (LAN) 21 and similarly Local Area Network (LAN) 21 may be located a substantial distance from Local Area Network (LAN) 24. That is, Local Area Network (LAN) 24 may be located in California while Local Area Network (LAN) 21 may be located within Texas and mainframe computer 27 may be located in New York.

This invention utilizes public and private key pairs for each party entity in the sealed transaction. A public and private key pair is a unique association of key values wherein one key can encrypt information and the other can decrypt. For example, the

AUS9-2000-0851

10

public key can encrypt the data and the corresponding private key can decrypt the encrypted data. Public and private keys are used for signing and sending encrypted messages. A public key is typically made available to users on the Internet within a certificate stored in a publicly assessable LDAP directory. The associated private key is kept in confidence by the entity, such as the person or corporation who owns the key pair.

Sealing the transaction involved signing the transaction with the sender's private key and encrypting the transaction with the recipient's public key. This process can also involve secret keys to help encrypt and decrypt bulk data. The sealed transaction is available to the recipient involved in the transaction, but not to unintended recipients. A sealed transaction meets the authentication, integrity, and confidentiality security requirements. This invention uses public key technologies to bring together the two different types of protocols and is applicable to current standards, as well as general synchronous and asynchronous transmissions.

Referring to Fig. 3, the message transmission using the present invention starts from a client 40. If the message or transaction starts from a client to a server using a synchronous protocol, this design will package the original transaction within a Public Key Cryptology Standard (PKCS) SignedData object 41 (The preferred PKCS being standard number 7). The SignedData object, as shown in Fig. 4 is a data structure with various fields that can link to other data structures. The SignedData object consists of the original message 50, signing certificate 51, signature bytes 52, and signed attributes 53. The signed attributes, shown in Figures 4 and 5, consist of the content or data type of the message 54, message digest 55, and signing time 56. Referring to Fig. 3, the SignedData object is sent to a receiver entity 42. Recipients of a SignedData object can verify the message and signature 43 using the sender's public key to ensure the authenticity of the sender and integrity of the message. The Recipient has to determine if the sender already has an existing message, which means the sender was a receiver of a previous transaction 44. If the recipient does not need to forward the message to another entity 46, the message transmission ends. When the sender does have an existing message, Recipient has to decide whether to continue to send the message or is this Recipient the final destination 45. If the message requires that the sender needs to continue sending the message on to another entity 45, the sender can generate add additional information 47 to

AUS9-2000-0851

11

the message. This additional information can include information such as an update to the transaction within a new SignedData object 48. The new SignedData object will have a new message (potentially), certificate of the current sender, signature bytes generated from the current sender's private key, and new signed attributes. The signed attributes will consist of the data type of the new message, the message digest of the new message, the signing time, and the message digest of the old message. This last signed attribute is known as the related message digest. Since message digest values are unique to the message for which the digest was generated, the related message digest attribute value allows the final or interim recipients to link the messages together to form the original chain of messages and determine the initial sender. (Optionally, the recipient can also use the signing time attribute to assist in the chronological ordering of the messages. However, there may be time drift between the entities that may make the signing time values unreliable.) All of the SignedData objects are grouped together and signed by the current sender to ensue that the receiver can trust all the internal SignedData objects by verifying the signature of all the data. After the determination that the message requires further transmission 49, the message packet is sent to the next entity and the protocol starts over in step 43.

The information from each entity during the transmission is kept in the final SignedData object. From this final SignedData object, one can trace all events in the transmission. For example, if the final recipient receives a message with the three SignedData objects, the recipient can verify the authenticity and integrity of the three SignedData objects since they would have been received within an encompassing SignedData object. The final recipient can determine the chains of messages by linking together the related message digest signed attribute value within one SignedData to a message digest signed attribute value of another SignedData object. (The first message will typically have the same related message digest and message digest signed attribute values). The recipient can check the authenticity and integrity of each SignedData object by verifying the certificate and using the certificate's public key to verify the signature of the message. From the chain of messages, the final recipient can understand how the message was changed each step of the way. Lastly, the final recipient knows the initial

[illegible]

AUS9-2000-0851

13

can show all transactions that occurred and all modifications to the message content. This record will assist in preserving the authenticity and integrity of the message during the transmission.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those skilled in the art will appreciate that the processes of the present invention are capable of being distributed in the form of instructions in a computer readable medium and a variety of other forms, regardless of the particular type of medium used to carry out the distribution. Examples of computer readable media include media such as EPROM, ROM, tape, paper, floppy disc, hard disk drive, RAM, and CD-ROMs and transmission-type of media, such as digital and analog communications links.

Having thus described the invention, what we claim as new and desire to secure by Letters Patent is set forth in the following claims.

15